(54) DESIGNING A 3D MODELED OBJECT VIA USER-INTERACTION

ENTWURF EINES 3D-MODELLIERTEN OBJEKTS ÜBER EINE BENUTZERINTERAKTION

CONCEPTION D'UN OBJET MODÉLISÉ 3D PAR L'INTERMÉDIAIRE D'UNE INTERACTION DE
L'UTILISATEUR

(56) References cited:
     EP-A1- 3 396 603

     • TAN QINGYANG ET AL: "Variational
       Autoencoders for Deforming 3D Mesh Models",
       2018 IEEE/CVF CONFERENCE ON COMPUTER
       VISION AND PATTERN RECOGNITION, IEEE, 18
       June 2018 (2018-06-18), pages 5841 - 5850,
       XP033473499, DOI: 10.1109/CVPR.2018.00612

     • QINGYANG TAN ET AL: "Mesh-based
       Autoencoders for Localized Deformation
       Component Analysis", ARXIV.ORG, CORNELL
       UNIVERSITY LIBRARY, 201 OLIN LIBRARY
       CORNELL UNIVERSITY ITHACA, NY 14853, 13
       September 2017 (2017-09-13), XP080820792

     • JERRY LIU ET AL: "Interactive 3D Modeling with
       a Generative Adversarial Network", ARXIV.ORG,
       CORNELL UNIVERSITY LIBRARY, 201 OLIN
       LIBRARY CORNELL UNIVERSITY ITHACA, NY
       14853, 16 June 2017 (2017-06-16), XP080770297,
       DOI: 10.1109/3DV.2017.00024

     • JUN-YAN ZHU ET AL: "Generative Visual
       Manipulation on the Natural Image Manifold",
       ARXIV.ORG, CORNELL UNIVERSITY LIBRARY,
       201 OLIN LIBRARY CORNELL UNIVERSITY
       ITHACA, NY 14853, 12 September 2016
       (2016-09-12), XP080996244

     • YUMER M ERSIN ET AL: "Learning Semantic
       Deformation Flows with 3D Convolutional
       Networks", 17 September 2016, PROC.
       INT.CONF. ADV. BIOMETRICS (ICB); [LECTURE
       NOTES IN COMPUTER SCIENCE; LECT.NOTES
       COMPUTER], SPRINGER, BERLIN,
       HEIDELBERG, PAGE(S) 294 - 311, ISBN:
       978-3-642-17318-9, XP047355304

EP 3 671 660 B1

Description

FIELD OF THE INVENTION

[0001] The invention relates to the field of computer programs and systems, and more specifically to a method, system and program for designing a 3D modeled object via user-interaction.

BACKGROUND

[0002] A number of systems and programs are offered on the market for the design, the engineering and the manufacturing of objects. CAD is an acronym for Computer-Aided Design, e.g. it relates to software solutions for designing an object. CAE is an acronym for Computer-Aided Engineering, e.g. it relates to software solutions for simulating the physical behavior of a future product. CAM is an acronym for Computer-Aided Manufacturing, e.g. it relates to software solutions for defining manufacturing processes and operations. In such computer-aided design systems, the graphical user interface plays an important role as regards the efficiency of the technique. These techniques may be embedded within Product Lifecycle Management (PLM) systems. PLM refers to a business strategy that helps companies to share product data, apply common processes, and leverage corporate knowledge for the development of products from conception to the end of their life, across the concept of extended enterprise. The PLM solutions provided by Dassault Systèmes (under the trademarks CATIA, ENOVIA and DELMIA) provide an Engineering Hub, which organizes product engineering knowledge, a Manufacturing Hub, which manages manufacturing engineering knowledge, and an Enterprise Hub which enables enterprise integrations and connections into both the Engineering and Manufacturing Hubs. All together the system delivers an open object model linking products, processes, resources to enable dynamic, knowledge-based product creation and decision support that drives optimized product definition, manufacturing preparation, production and service.

[0003] In this context and other contexts, machine-learning and in particular autoencoder and/or manifold learning is gaining wide importance.

[0004] The following papers relate to this field and are referred to hereunder:

[1]: "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", P. Vincent, H. Larcohelle, I. Lajoie, Y. Bengio, P. Manzagol, in The Journal of Machine Learning Research, 2010.
[2]: "Reducing the Dimensionality of Data with Neural Networks", G.E. Hinton, R.R. Salakhutdinov, in Science, 2006.
[3]: "Learning Deep Architectures for AI", Y. Bengio, in Foundations and Trends in Machine Learning, 2009.
[4]: "Generative Adversarial Nets", I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, in Advances in Neural Information Processing Systems, 2014.
[5]: "Generative Visual Manipulation on the Natural Image Manifold", J-Y Zhu, P. Krähenbühl, E. Shechtman, A. A. Efros, in ECCV 2016.
[6]: "Neural Photo Editing with Introspective Adversarial Networks", A. Brock, T. Lim, J.M. Ritchie, in ICLR 2017.

[0005] Given a dataset representing samples from a same class, for example a dataset of images of chairs, or a dataset of 3D models of cars, autoencoders (explained in [1,2]) aims at learning a mapping between the original input space of this dataset and a low-dimensional space, often called latent space. Moreover, autoencoders also learn a reverse mapping from the latent space to the original input space. They may be used to do realistic non-linear interpolation in the original spaces, extract meaningful features in the data, compress the data into a compact representation, etc.

[0006] Paper [4] introduces a generative adversarial network, a model which learns a decoder, but also sets a probabilistic prior on the latent space to learn the distribution of the data. To do so, they train the decoder in order to fool a discriminator, whereas the discriminator is trained to classify between real data and synthetized data. One can also learn an encoder to project the data onto the latent space, using a reconstruction error like autoencoders.

[0007] Paper [5] learns a generative adversarial network on images. The modifications are finally retargeted onto the original image with optical flow. Paper [6] also proposes an edition system onto images, by learning an introspective adversarial network, an hybridization between a variational autoencoder and a generative adversarial network, but they retarget the user modifications with masking techniques instead of optical flow.

[0008] However, the use of machine-learning techniques in the context of (e.g. 3D) design remains an uncommon and relatively unknown approach.

[0009] Also known is EP3396603A1, which discloses a computer-implemented method for learning an autoencoder. The method comprises providing a dataset of images. Each image includes a respective object representation. The method also comprises learning the autoencoder based on the dataset. The learning includes minimization of a reconstruction loss. The reconstruction loss includes a term that penalizes a distance for each respective image. The

penalized distance is between the result of applying the autoencoder to the respective image and the set of results of applying at least part of a group of transformations to the object representation of the respective image.

[0010] Also known is Q. Tan et. al. "Variational Autoencoders for Deforming 3D Mesh Models", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. This document discloses a framework called mesh variational autoencoders (mesh VAE), to explore the probabilistic latent space of 3D surfaces.

[0011] Also known is E. Yumer et. al. "Learning Semantic Deformation Flows with 3D Convolutional Networks", Springer International Publishing AG 2016, B. Leibe et al. (Eds.): ECCV 2016, Part VI, LNCS 9910, pp. 294-311, 2016, DOI: 10.1007 /978-3-319-46466-4_18. This document discloses an end-to-end solution to the process of shape deformation using a volumetric Convolutional Neural Network (CNN) that learns deformation flows in 3D. The network architectures take the voxelized representation of the shape and a semantic deformation intention as input and generate a deformation flow at the output.

[0012] Also known is J. Liu et. al. "Interactive 3D Modeling with a Generative Adversarial Network", arXiv:1706.05170v1 16 Jun 2017. This document discloses the idea of using a generative adversarial network (GAN) to assist a novice user in designing real-world shapes with a simple interface. The user edits a voxel grid with a painting interface (like Mine craft). Yet, at any time, he/she can execute a SNAP command, which projects the current voxel grid onto a latent shape manifold with a learned projection operator and then generates a similar, but more realistic, shape using a learned generator network. Then the user can edit the resulting shape and snap again until he/ she is satisfied with the result.

[0013] Also known is Q. Tan et. al. "Mesh-based Autoencoders for Localized Deformation Component Analysis", arXiv:1709.04304v1 13 Sep 2017. This document discloses a mesh-based autoencoder architecture that is able to cope with meshes with irregular topology. Sparse regularization is introduced in this framework, which along with convolutional operations, helps localize deformations.

[0014] Also known is J-Y. Zhu et. al., "Generative Visual Manipulation on the Natural Image Manifold", ar-Xiv:1609.03552V3 16 Dec 2018. This document discloses the learning of the natural image manifold directly from data using a generative adversarial neural network. Then a class of image editing operations is defined, and their output is constrained to lie on that learned manifold at all times. The model automatically adjusts the output keeping all edits as realistic as possible. All the manipulations are expressed in terms of constrained optimization and are applied in near-real time.

[0015] Within this context, there is still a need for an improved method for designing a 3D modeled object via user-interaction.

SUMMARY OF THE INVENTION

[0016] It is therefore provided a computer-implemented method according to appended claim 1.

[0017] Such a method constitutes an improved method for designing a 3D modeled object via user-interaction.

[0018] Notably, the method makes use of a machine-learnt decoder in the context of designing a 3D modeled object via user-interaction.

[0019] Furthermore, the user defines a deformation constraint for a part of the 3D modeled object. In other words, the user provides a design intent of which purpose is to deform the 3D modeled object. The method then determines the optimal latent vector, which is a vector minimizing an energy that penalizes non-respect of the deformation constraint on the latent vector space. In other words, the determined optimal latent vector is a vector of the latent space tending to respect the deformation constraint. In examples, the optimal latent vector respects (e.g. optimally) the deformation constraint. The optimal latent vector is then decoded into an element of the 3D modeled object space. In other words, the method finds a 3D modeled object which tends to respect (*e.g.* respects, *e.g.* optimally) the deformation constraint provided by the user. Hence, the method allows to find a 3D modeled object which tends to respect (e.g. respects) the user's design intent.

[0020] Moreover, the determining of the optimal latent vector and the applying of the decoder to the optimal latent vector may be carried out automatically, thus without involving user actions, as a result of the providing of the deformation constraint by the user. Thus, not only does the method output a 3D modeled object tending to respect the user's design intent, but the method does so with few user-machine interactions. The method is thus efficient and ergonomic.

[0021] Also, the decoder may output only plausible 3D modeled objects because the decoder may have been machine-learnt to do so. For example, the 3D modeled object space may be a space of plausible 3D modeled objects. In other words, the method may output a plausible 3D modeled object tending to respect the user's design intent in an ergonomic and efficient way.

[0022] The method may comprise one or more of the features recited in the appended claims which depend on appended claim 1.

[0023] It is further provided a computer program comprising instructions for performing the method.

[0024] It is further provided a computer readable storage medium having recorded thereon the computer program.

[0025] It is further provided a system comprising a processor coupled to a memory and a graphical user interface, the

memory having recorded thereon the computer program.

BRIEF DESCRIPTION OF THE DRAWINGS

5   [0026]   Embodiments of the invention will now be described, by way of non-limiting example, and in reference to the accompanying drawings, where:

- FIG. 1 shows a flowchart of an example of the method;
- FIG.s 2 to 13 illustrate the method;
10   - FIG. 14 shows an example of a graphical user interface of the system; and
- FIG. 15 shows an example of the system.

DETAILED DESCRIPTION OF THE INVENTION

15   [0027]   With reference to the flowchart of FIG. 1, it is proposed a computer-implemented method for designing a 3D modeled object via user-interaction. The method comprises providing S10 the 3D modeled object and a machine-learnt decoder. The machine-learnt decoder is a differentiable function taking values in a latent space and outputting values in a 3D modeled object space. The method further comprises defining S30, by a user, a deformation constraint for a part of the 3D modeled object. The method further comprises determining S40 an optimal vector. The optimal vector minimizes an
20   energy. The energy explores latent vectors. The energy comprises a term which penalizes, for each explored latent vector, non-respect of the deformation constraint by the result of applying the decoder to the explored latent vector. The method further comprises applying S50 the decoder to the optimal latent vector.
[0028]   The method is computer-implemented. This means that steps (or substantially all the steps) of the method are executed by at least one computer, or any system alike. Thus, steps of the method are performed by the computer, possibly
25   fully automatically, or, semi-automatically. In examples, the triggering of at least some of the steps of the method may be performed through user-computer interaction. The level of user-computer interaction required may depend on the level of automatism foreseen and put in balance with the need to implement user's wishes. In examples, this level may be user-defined and/or pre-defined.
[0029]   A typical example of computer-implementation of a method is to perform the method with a system adapted for
30   this purpose. The system may comprise a processor coupled to a memory and a graphical user interface (GUI), the memory having recorded thereon a computer program comprising instructions for performing the method. The memory may also store a database. The memory is any hardware adapted for such storage, possibly comprising several physical distinct parts (e.g. one for the program, and possibly one for the database).
[0030]   The method generally manipulates modeled objects. A modeled object is any object defined by data stored e.g. in
35   the database. By extension, the expression "modeled object" designates the data itself. According to the type of the system, the modeled objects may be defined by different kinds of data. The system may indeed be any combination of a CAD system, a CAE system, a CAM system, a PDM system and/or a PLM system. In those different systems, modeled objects are defined by corresponding data. One may accordingly speak of CAD object, PLM object, PDM object, CAE object, CAM object, CAD data, PLM data, PDM data, CAM data, CAE data. However, these systems are not exclusive one
40   of the other, as a modeled object may be defined by data corresponding to any combination of these systems. A system may thus well be both a CAD and PLM system, as will be apparent from the definitions of such systems provided below.
[0031]   By CAD system, it is additionally meant any system adapted at least for designing a modeled object on the basis of a graphical representation of the modeled object, such as CATIA. In this case, the data defining a modeled object comprise data allowing the representation of the modeled object. A CAD system may for example provide a representation
45   of CAD modeled objects using edges or lines, in certain cases with faces or surfaces. Lines, edges, or surfaces may be represented in various manners, e.g. non-uniform rational B-splines (NURBS). Specifically, a CAD file contains specifications, from which geometry may be generated, which in turn allows for a representation to be generated. Specifications of a modeled object may be stored in a single CAD file or multiple ones. The typical size of a file representing a modeled object in a CAD system is in the range of one Megabyte per part. And a modeled object may typically be an assembly of thousands
50   of parts.
[0032]   The concept of "3D modeled object" involved in the method is now discussed.
[0033]   In the context of CAD, a modeled object may typically be a 3D modeled object, e.g. representing a product such as a part or an assembly of parts, or possibly an assembly of products. By "3D modeled object", it is meant any object which is modeled by data allowing its 3D representation. A 3D representation allows the viewing of the part from all angles. For
55   example, a 3D modeled object, when 3D represented, may be handled and turned around any of its axes, or around any axis in the screen on which the representation is displayed. This notably excludes 2D icons, which are not 3D modeled. The display of a 3D representation facilitates design (i.e. increases the speed at which designers statistically accomplish their task). This speeds up the manufacturing process in the industry, as the design of the products is part of the manufacturing

process.

[0034] In the context of the method, the 3D modeled object may represent the geometry of a product to be manufactured in the real world subsequent to the completion of its virtual design with for example a CAD software solution or CAD system, such as a (e.g. mechanical) part or assembly of parts (or equivalently an assembly of parts, as the assembly of parts may be seen as a part itself from the point of view of the method, or the method may be applied independently to each part of the assembly), or more generally any rigid body assembly (e.g. a mobile mechanism). A CAD software solution allows the design of products in various and unlimited industrial fields, including: aerospace, architecture, construction, consumer goods, high-tech devices, industrial equipment, transportation, marine, and/or offshore oil/gas production or transportation. The 3D modeled object designed by the method may thus represent an industrial product which may be any mechanical part, such as a part of a terrestrial vehicle (including e.g. car and light truck equipment, racing cars, motorcycles, truck and motor equipment, trucks and buses, trains), a part of an aerial vehicle (including e.g. airframe equipment, aerospace equipment, propulsion equipment, defense products, airline equipment, space equipment), a part of a naval vehicle (including e.g. navy equipment, commercial ships, offshore equipment, yachts and workboats, marine equipment), a general mechanical part (including e.g. industrial manufacturing machinery, heavy mobile machinery or equipment, installed equipment, industrial equipment product, fabricated metal product, tire manufacturing product), an electro-mechanical or electronic part (including e.g. consumer electronics, security and/or control and/or instrumentation products, computing and communication equipment, semiconductors, medical devices and equipment), a consumer good (including e.g. furniture, home and garden products, leisure goods, fashion products, hard goods retailers' products, soft goods retailers' products), a packaging (including e.g. food and beverage and tobacco, beauty and personal care, household product packaging).

[0035] In examples, any 3D modeled object designed by the method and/or involved in the method may represent a mechanical part which is one or a plausible combination of a molded part *(i.e.* a part manufactured by a molding manufacturing process), a machined part *(i.e.* a part manufactured by a machining manufacturing process), a drilled part *(i.e.* a part manufactured by a drilling manufacturing process), a turned part *(i.e.* a part manufactured by a turning manufacturing process), a forged part *(i.e.* a part manufactured by a forging manufacturing process), a stamped part *(i.e.* a part manufactured by a stamping manufacturing process) and/or a folded part *(i.e.* a part manufactured by a folding manufacturing process).

[0036] Any 3D modeled object designed by the method and/or involved in the method may be a plausible (e.g. realistic) 3D modeled object. A plausible 3D modeled object may designate a 3D modeled object representing a plausible (e.g. realistic) mechanical part. A plausible mechanical part may designate a mechanical part realistically manufacturable in a real-world industrial manufacturing process. A plausible mechanical part may refer to a mechanical part which respects all the constraints that must be respected in order to realistically manufacture the mechanical part in a real-world industrial manufacturing process. The constraints may comprise one or any plausible combination of the following: mechanical constraints (e.g. constraints resulting from the laws of classic mechanics), functional constraints (e.g. constraints relative to one or more mechanical functions to be performed by the mechanical part once manufactured), manufacturing constraints (e.g. constraints pertaining to the ability of applying one or more manufacturing tools to the mechanical part during one or more manufacturing processes for manufacturing the mechanical part), structural constraints (e.g. constraints pertaining to the strength and/or resistance of the mechanical part) and/or assembly constraints (*e.g.* constraints defining how the mechanical part can be assembled with one or more other mechanical parts).

[0037] "Designing a 3D modeled object" designates any action or series of actions which is at least part of a process of elaborating a 3D modeled object. Thus, the method may comprise creating the 3D modeled object from scratch. Alternatively, the method may comprise providing a 3D modeled object previously created, and then modifying the 3D modeled object.

[0038] The method may be included in a manufacturing process, which may comprise, after performing the method, producing a physical product corresponding to the modeled object. In any case, the modeled object designed by the method may represent a manufacturing object. The modeled object may thus be a modeled solid (i.e. a modeled object that represents a solid). The manufacturing object may be a product, such as a part, or an assembly of parts. Because the method improves the design of the modeled object, the method also improves the manufacturing of a product and thus increases productivity of the manufacturing process.

[0039] The providing S10 of the 3D modeled object is now discussed.

[0040] The providing of the 3D modeled object may comprise displaying the 3D modeled object on a display of a computer such as a CAD system. The providing of the 3D modeled object may result from an action of a user, *e.g.* an interaction between the user and the computer. In examples, the 3D modeled object may have been designed by a another user on another computer and optionally stored in a memory and/or sent (*e.g.* through a network) to the computer. The providing of the 3D modeled object may comprise retrieving the 3D modeled object from a memory. In examples, the user and the another user are different, and the computer and the another computer are different. In these examples, the computer and the another computer may be connected by a network. In examples, the user and the another user are the same, and the computer and the another computer are the same.

[0041] The 3D modeled object may be provided S10 as a mesh or as a point cloud. Alternatively, the providing S10 may comprise providing the 3D modeled object in any format (e.g. as a CAD file) and meshing the 3D modeled object and/or determining (e.g. inferring, e.g. extracting, e.g. computing), from the (e.g. meshed) 3D modeled object, a point cloud representing the 3D modeled object.

[0042] The machine-learnt decoder is now discussed.

[0043] The machine-learnt decoder is a differentiable function. The function is machine-learnt. The function takes is values in a latent space. The latent space is a vector space. The function outputs values in a 3D modeled object space, i.e. a space comprising (e.g. made of) 3D modeled objects. In other words, applying the function to vectors of the latent space (hereinafter referred to as "latent vectors") outputs 3D modeled objects. In other words, the image of each latent vector by the function is a 3D modeled object of the 3D modeled object space. The image of the latent space by the function may be the 3D modeled object space or a strict subset of the 3D modeled object space. Decoding a latent vector means applying the decoder to the latent vector. The result of the decoding may thus be referred to as "the decoded latent vector". In examples, the latent space features a probabilistic prior. In other words, a probabilistic prior, which may be denoted $p(z)$ in the following, may be given over the latent space, z being a notation for a vector of the latent space. In these examples, the prior may be an *a priori* probability distribution expressing how the latent vectors are distributed over the latent space. A *priori* means that the probability distribution may be determined (e.g. computed, e.g. established) without having complete knowledge of the latent space. For example, the prior over the latent space may be inferred (e.g. deduced) from an existing (e.g. probability) distribution of the 3D modeled objects over the 3D modeled object space, such that the prior expresses a distribution of the latent vectors over the latent space that (e.g. substantially) corresponds to the distribution of the 3D modeled objects over the 3D modeled object space.

[0044] The machine-learnt decoder may be the decoder of an autoencoder. In the context of the method, an autoencoder may be defined as the composition of two feedforward deep neural networks (see [3])

$$f_w : \mathbb{R}^m \to \mathbb{R}^p \quad \text{and} \quad g_{w'} : \mathbb{R}^p \to \mathbb{R}^m$$

, parameterized by the weights w and w', where $p \ll m$. f_w is the encoder, and $g_{w'}$ the decoder. $\mathbb{R}^p$ is the latent space, i.e. the vector space where the encoder f_w outputs its values and where the decoder $g_{w'}$ takes its values. p is the dimension of the latent space. $\mathbb{R}^m$ is the space where the encoder f_w takes its values and where the decoder $g_{w'}$ outputs its values. $\mathbb{R}^m$ may be referred to as "the original input space". m is the dimension of the original input space. In the case of the machined-learnt decoder provided by the method, the original input space may be the 3D modeled object space. Alternatively, the 3D modeled object space may be a subset of the original input space. In other words, the encoder $f_w(x)$ takes its values in the original input space, but the image of the latent space by the decoder $g_{w'}$, i.e. the 3D modeled object space in the context of the method, may be a subset of the original input space. A vector $z = f_w(x)$ may be referred to as "a latent vector" or "a hidden vector". The autoencoder $g_{w'} \circ f_w(x)$ may also be referred to as "a reconstruction". The reconstruction takes as input a first element of the original input space (for example, a first 3D modeled object of the 3D modeled object space), maps it onto a latent vector of the latent vector space, and then reverse the mapping by outputting a second element of the original input space (for example, a second 3D modeled object of the 3D modeled object space) from the latent vector. The second element may be referred to as "a reconstruction" of the first element. In examples it means that the second element represents an approximation of the first element. In examples, if x is a 3D modeled object, the 3D modeled object $g_{w'} \circ f_w(x)$ may be referred to as the reconstructed 3D modeled object. x may also be referred to as the input and $g_{w'} \circ f_w(x)$ as the reconstructed input.. In the context of the method, the encoder f_w, and the decoder g_w may be machined-learnt (e.g. separately or simultaneously), e.g. by minimizing a reconstruction error. Encoding an object means applying the encoder to the object. The result of the encoding may thus be referred to as "the encoded object". Decoding a latent vector means applying the decoder to the latent vector. The result of the decoding may thus be referred to as "the decoded latent vector".

[0045] The providing S10 of the machine-learnt decoder is now discussed.

[0046] The providing S10 of the machine-learnt decoder may comprise retrieving the machine-learnt decoder from a database. In other words, the machine-learnt decoder may have been previously learnt and stored in a database for later use. In examples, the providing S10 of the machine-learnt decoder may be carried out by a user which accesses the decoder in a database. Additionally or alternatively, the providing S10 may comprise learning the decoder from scratch and making it available for use by the method.

[0047] The 3D modeled object space is now discussed.

[0048] The 3D modeled object space generally refers to a space made of 3D modeled objects. The 3D modeled object space may be included in a vector space. In examples, the 3D modeled object space is a manifold, for example a connected manifold, of a vector space. In examples, the 3D modeled object space is made of 3D modeled objects of a same class of 3D modeled objects. A class of 3D modeled objects may correspond to a connected manifold of a vector space. Additionally or alternatively, a class of 3D modeled objects may designate a set of 3D modeled objects which has a the following property: any first 3D modeled object of the set is similar to, e.g. has a shape similar to, at least a second 3D modeled object of the set.

[0049] In examples, the class of 3D modeled object is made of 3D modeled object representing respective mechanical parts, the mechanical parts relative (i.e. corresponding) to the class all respecting any one or any combination of the following conditions:

- the mechanical parts relative to the class are all manufactured in the same manufacturing process or in the same combination of manufacturing processes;
- the mechanical parts relative to the class are all plausible mechanical parts;
- the mechanical parts relative to the class are all from a same field of technology and/or industry;
- the mechanical parts relative to the class all perform a same mechanical function;
- the mechanical parts relative to the class each being represented by a 3D modeled object which has a shape similar to at least one other 3D modeled object of the class (thus representing another mechanical part relative to the class); and/or
- the mechanical parts relative to the class all obey to *(e.g.* satisfy, *e.g.* respect e.g. verify) same mechanical constraints, functional constraints, manufacturing constraints, structural constraints and/or assembly constraints.

[0050] In examples, the provided 3D modeled object and (e.g. substantially) all the 3D modeled objects of the 3D modeled object space are 3D modeled objects of a same class of 3D modeled objects. In these examples, the decoder may have been machine-learnt based on a dataset of 3D modeled objects of said same class. For example, the decoder may have been machine-learnt to map any latent vector onto a 3D modeled object of said same class. In examples, the decoder may be the decoder of an autoencoder that further comprises an encoder, the decoder and the encoder having both being machine-learnt based on a dataset of 3D modeled objects of said same class. In these examples, the encoder may have been machine-learnt to map any 3D object of said same class onto a latent vector of the latent space. In these examples, the decoder may have been machine-learnt to map any latent vector onto a 3D modeled object of said same class. In examples, all modeled object of the 3D modeled object space are plausible 3D modeled objects.

[0051] FIG. 2 shows a class of 3D modeled objects forming an example of the set 20 of 3D modeled object of the method. The 3D modeled objects are chairs. The chairs form a class of 3D modeled object because they all perform the same mechanical function, which is supporting a weight (e.g. of a human being). Performing this mechanical function also means that the chairs all obey to same mechanical constraints, functional constraints and structural constraints. The chairs of FIG. 2 are plausible mechanical parts, because they respect mechanical constraints, functional constraints, manufacturing constraints and structural constraints, thereby allowing their manufacturing in the real-world by one or more manufacturing process.

[0052] FIG. 3 shows a class of 3D modeled objects forming an example of the set 30 of 3D modeled object of the method. The 3D modeled objects are chairs with four legs (each). The chairs with four legs form a class of 3D modeled object because they all perform the same mechanical function, which is supporting a weight (*e.g.* of a human being), and because performing this mechanical function with four legs also means that the chairs all obey to same mechanical constraints, functional constraints and structural constraints. The chairs of FIG. 3 are plausible mechanical parts, because they respect mechanical constraints, functional constraints, manufacturing constraints and structural constraints, thereby allowing their manufacturing in the real-world by one or more manufacturing process.

[0053] The 3D modeled object space may be a space made of discrete representations of 3D modeled objects, such as meshes and/or point clouds. In other words, the 3D modeled object space is made of respective discrete representations (such as meshes and/or point clouds) of 3D modeled objects of the space. In such cases, the elements of the 3D modeled object space may still be referred to as the 3D modeled objects themselves, and not their respective discrete representations, for the sake of simplicity.

[0054] In examples, the method comprises projecting S20 the 3D modeled object onto the latent space. In these examples, the machine-learnt decoder may be the decoder of a machine-learnt autoencoder also comprising a machine-learnt encoder. Projecting S20 the 3D modeled object may consist in applying the encoder to the 3D modeled object, the applying of the encoder to the 3D modeled object resulting in a latent vector. The latent vector may be called the projection of the 3D modeled object on the latent space. Alternatively, the providing S10 of the 3D modeled object may comprise providing, besides the 3D modeled object, the latent vector which is the result of applying the encoder to the 3D modeled object. In any case, the projecting S20 may result in that a latent vector which is the projection of the 3D modeled object onto the latent space is made available for further use in the method, as discussed herein after.

[0055] The part of the 3D modeled object is now discussed.

[0056] In examples, any 3D modeled objects designed by the method or involved in the method may be divided into parts. For a 3D modeled object representing a mechanical part, a part of the 3D modeled object may designate a layout of material of the mechanical part, the layout of material performing a mechanical function. In other words, the 3D modeled object representing the mechanical part may be divided into parts each representing a respective layout of material of the mechanical parts, the respective layout of material performing a respective mechanical function. In the context of the invention, a function performed by a layout of material of a mechanical part may be one or a plausible combination of a

supporting function, a strengthening function, a resistance function, a connecting function between other layouts of material, a mass reduction function, a space reservation function, a fixture function, a tightness function, an adjustment function, a positioning function, a mechanical joint function, a cooling function, a space reservation function, a revolute or cylindrical mechanical joint function, an assembly function, a stiffening function, a positioning pin function, a revolute or cylindrical mechanical joint function and/or a support for all machined and drilled protrusions function.

[0057] For example, a leg of a chair, such as a leg of any one of the chairs represented in FIG. 2 or FIG. 3, is a layout of material of the chair performing a supporting function. In other words, the leg of the chair is involved (*e.g.* as well as other legs) in performing the function of supporting the weight of further material forming the chair as well as the weight of a human sitting on the chair.

[0058] Another example is a (*e.g.* bent) chair back, such as a chair back of any one of the chairs represented in FIG. 2 or FIG. 3. The chair back is a layout of material of the chair performing a supporting function. In other words, the chair back is involved in performing the function of supporting the back of a human sitting on the chair.

[0059] Another example is a chair seat, such as a chair seat of any one of the chairs represented in FIG. 2 or FIG. 3. The chair seat is a layout of material of the chair performing a supporting function. In other words, the chair seat is involved in performing the function of supporting the weight of further material forming the chair as well as the weight of a human sitting on the chair. The chair seat also performs a connecting function, since it connects other parts of the chair such as the chair back and/or the chair legs.

[0060] Other examples of 3D modeled objects are now discussed.

[0061] Any 3D modeled object involved in the method may represent a car. In such a case, the part of the 3D modeled object may represent a wheel of the car (performing a supporting function as well as the function of rolling and/or turning) or a part of the car body (performing a connecting function and/or a resistance function and/or a strengthening function) such as a door, a hood, or a deck lid. Cars may form a class of 3D modeled objects. Alternatively or additionally, any 3D modeled object involved in the method may represent a plane. In such a case, the part of the 3D modeled object may represent a wing of the plane (performing a function of producing lift while moving through a fluid) or an engine of the plane (performing a function of producing force to move the plane) or the fuselage of the plane (performing a connecting function and/or a resistance function and/or a strengthening function). Planes may form a class of 3D modeled objects.

[0062] The defining S30 of the deformation constraint for the part of the 3D modeled object is now discussed.

[0063] The deformation constrain is defined S30 by a user, i.e. may result from a user action. The deformation constraint may designate a set of (e.g. geometric) specifications that determine a deformation (*e.g.* a transformation, *e.g.* a displacement) of a geometrical shape of the part of the 3D modeled object into another geometrical shape (which may be referred to as "the new geometrical shape" in the following). In other words, the deformation may not be directly defined by the user, but the user action rather results in specifications forcing the deformation. In examples, it means that the user provides (*e.g.* declares, *e.g.* specifies), *e.g.* by using a keyboard and/or a touch and/or a haptic device, the set of specifications. Alternatively, the set of specifications may be (*e.g.* automatically) retrieved from a user action on (*e.g.* the part of) the 3D modeled object. Alternatively, the set of specifications may be (*e.g.* automatically) retrieved from a drawing of the user on a screen comprising the 3D modeled object. The deformation defined by the deformation constraint may thus deform the 3D modeled object into a deformed 3D modeled object by deforming the geometrical shape of the part into the new geometrical shape.

[0064] The defining S30 of the deformation constraint results from a graphical-interaction of the user with the part of the 3D modeled object. The graphical-interaction may comprise clicking on the 3D modeled object, *e.g.* with a haptic device such as a mouse, or touching the 3D modeled object with an appendage such as a finger. The graphical interaction may further comprise moving the clicked on or touched 3D modeled object, for example by performing a drag and drop operation. For example, the user selects one or more points on the part of the 3D modeled object, *e.g.* by clicking on the one or more points with a haptic device such as a mouse and moves them by a drag and drop operation (*e.g.* moves them one by one by one or through a one-shot drag and drop operation), as known in the art. The respective one or more locations of the moved one or more points determine (*e.g.* form, *e.g.* result in) the new geometrical shape of the part of the 3D modeled object. In other words, the 3D modeled object has a given geometrical shape at the providing S10, and the user defines the new geometrical shape by moving the points. The deformation of the given geometrical shape into the new geometrical shape is thus constrained by the respective one or more locations of the one or more points moved by the user. In other words, moving the one or more points defines the new geometrical shape of the part and the new geometrical shape of the part defines the deformation constraint.

[0065] Alternatively, the user may define the deformation constraint by drawing (*e.g.* by using a haptic device) a 2D drawing (*e.g.* on a display) which represents the new geometrical shape for the part of the 3D modeled object, which has a given geometrical shape at the providing S10. The deformation of the given geometrical shape into the new geometrical shape is thus constrained by the drawing of the user. In other words, the drawing of the user defines the new geometrical shape of the part and the new geometrical shape of the part defines the deformation constraint.

[0066] The determining S40 of the optimal latent vector is now discussed.

[0067] The optimal latent vector minimizes an energy. Thus, the determining S40 of the latent vector may consist in

minimizing the energy, the optimal latent vector being the result of the minimizing. The energy explores latent vectors. In examples, this means that the energy is a function of at least a variable representing latent vectors, *i.e.* the at least one variable belongs to the latent space or to a subspace of the latent space. Minimizing the energy may be carried out iteratively, *i.e.* by iteratively visiting latent vectors of at least a subspace of the latent space until (*e.g.* an approximation of) the latent vector minimizing the energy (*i.e.* the optimal latent vector) is found. Minimizing the energy may be carried out automatically upon the defining S30 the deformation constraint. Minimizing the energy may be carried out by running any minimization algorithm, such as an iteratively reweighted least squares algorithm.

[0068] The energy comprises a term which, penalizes, for each explored latent vector, non-respect of the deformation constraint by the result of applying the decoder to the explored latent vector. In examples where the energy is a function of at least a variable representing latent vectors, the term may depend on the at least variable and may provide a measure (*e.g.* a quantification) of the non-respect of the deformation constraint by the result of applying the decoder to the at least variable. In these examples, penalizing may mean that the term is an increasing function of the largeness of the measure.

[0069] The method thus determines an optimal latent vector by exploring latent vectors and verifying that the decoded latent vectors (*i.e.* the results of applying the decoder to the explored latent vectors) respect or do not respect the deformation constraint. Non-respect is penalized, such that the decoder optimal latent vector tends to respect (*e.g.* respects) the deformation constraint. The decoded latent vector may be a plausible 3D modeled object, such that, in this case, the method outputs a 3D modeled object which is plausible, and which tends to respect (*e.g.* respects) de deformation constraint. All decoded explored latent vectors may also be plausible 3D modeled objects. In other words, the method may determine a plausible 3D modeled object which tends to respect (*e.g.* respects, *e.g.* respects best) the deformation constraint among a plurality of plausible 3D modeled objects. Moreover, the 3D modeled object space may be a class of 3D modeled object, such that, in this case, the method may determine a plausible 3D modeled object of a class which tends to respect (*e.g.* respects, *e.g.* respects best) the deformation constraint among a plurality of plausible 3D modeled objects of the class. Furthermore, the method may do so automatically, such that in these examples the method is efficient and ergonomic.

[0070] The deformation constraint defines a geometrical shape, and the term penalizes a disparity between the result of applying the decoder to the explored latent vector and the geometrical shape.

[0071] The deformation defined by the deformation constraint may deform the shape of the part of the 3D modeled object into a new geometrical shape of part of the 3D modeled object. As a result, the deformation may also deform the whole 3D modeled object into a new 3D modeled object (which may be also referred to as the deformed input) by deforming the geometrical shape of the part. In other words, the new 3D modeled object may differ from the provided 3D modeled object in that the geometrical shape of the part is deformed. The geometrical shape defined by the deformation constraint may thus designate the new 3D modeled object. In examples where the decoder is the decoder of an autoencoder, the new 3D modeled object may be computed from the reconstructed 3D modeled object. That is, the autoencoder is applied to the 3D modeled object, thereby outputting a reconstructed 3D modeled object, and the new 3D modeled object is computed by applying the deformation defined by the deformation constraint to the reconstructed 3D modeled object. Alternatively, the geometrical shape defined by the deformation constraint may designate the new geometrical shape of the part of the 3D modeled object.

[0072] The disparity between the result of applying the decoder to the explored latent vector and the geometrical shape may be a quantity relative to (*e.g.* that is a function of) a difference or an error between the result of applying the decoder to the explored latent vector and the geometrical shape defined by the deformation constraint. In other words, the geometrical shape being defined by the deformation constraint, the disparity may be a quantity which provides a measure (*e.g.* a quantification) of the non-respect of the deformation constraint by the result of applying the decoder to the explored latent vector. In these examples, penalizing may mean that the quantity is an increasing function of the largeness of the measure.

[0073] In examples where the 3D modeled objects involved in the method are plausible 3D modeled objects, the method thus visits plausible 3D modeled objects (*i.e.* the decoded explored latent vectors) and penalized disparities between visited plausible 3D modeled objects and the geometrical shape defined by the deformation constraint, until a 3D modeled object (the decoded optimal latent vector) for which the disparity is acceptable (*e.g.* is the less penalized) is determined. In other words, the method determines a plausible 3D modeled object of which geometrical shape is (*e.g.* best) compliant with the geometrical shape defined by the deformation constraint.

[0074] The disparity involves a distance between a derivative of the result of applying the decoder to the explored latent vector and the geometrical shape.

[0075] The derivative of the result of applying the decoder to the explored latent vector may be a mathematical quantity obtained from the result of applying the decoder to the explored latent vector, *e.g.* by applying one or more mathematical formulae. In other words, the derivative of the result of applying the decoder to the explored latent vector may be a function of the result of applying the decoder to the explored latent vector. The disparity involves the distance between a derivative of the result of applying the decoder to the explored latent vector and the geometrical shape. In examples, it means that the disparity is a function of the distance.

[0076] Using the distance between a derivative of the result of applying the decoder to the explored latent vector and the

geometrical shape is a simple and efficient way to measure (*e.g.* to quantify) non-respect of the deformation constraint. The method is thus efficient in these examples.

[0077] The disparity is a monotonic function of the distance.

[0078] The method thus measures efficiently, in these examples, non-respect of the deformation constraint: the more the deformation constraint is not respected, the larger the distance is, and the larger the disparity is. Conversely, the less the deformation constraint is not respected, the smaller the distance is, and the smaller the disparity is. The method is thus efficient.

[0079] The geometrical shape may be defined in a 3D space, in which case the derivative is the result of applying the decoder to the explored latent vector. In these examples, the geometrical shape may thus represent a new geometrical shape of the 3D modeled object, *i.e.* the geometrical shape of the 3D modeled object after the part of the 3D modeled object has undergone the deformation corresponding to the deformation constraint.

[0080] Alternatively, the geometrical shape may be defined in a 2D plane, in which case the derivative is a projection of the result of applying the decoder to the explored latent vector. In examples, the geometrical shape may thus be a 2D drawing of the user representing the new geometrical shape for the part of the 3D modeled object.

[0081] The 3D modeled object is a 3D mesh. The 3D modeled object space is a space made of 3D meshes. The machine-learnt decoder thus takes values in the latent space and outputs 3D meshes.

[0082] When the geometrical shape is defined in a 3D space and the derivative is the result of applying the decoder to the explored latent vector, the term is of the type:

$$\sum_{i=1}^{k}\left[\sum_{j=1}^{M}\rho\left(\|g(z)_{v_i} - g(z)_j\|\right)\|g(z)_j - y_j\|_2^2\right](1).$$

[0083] In formula (1), z is the explored latent vector, g the provided decoder, and $g(z)$ is the result of applying the decoder to the explored latent vector. $v_1, v_2, \ldots, v_k$ are the indices of k vertices of $g(z)$ representing the part of the 3D modeled object. In other words, the 3D modeled object is a 3D mesh, the part of the 3D modeled object is a sub mesh of the 3D mesh, $g(z)$ is also a 3D mesh (since the 3D modeled object space is a 3D mesh space), and $v_1, v_2, \ldots, v_k$ are the indices of k vertices of $g(z)$ representing the sub mesh of the 3D mesh. Representing may mean approximating, such that the sub mesh with indices $v_1, v_2, \ldots, v_k$ is the sub mesh of $g(z)$ which approximates the best (*e.g.* which is the closest to) the part of the 3D modeled object. M is a sampling size of the 3D modeled object, *e.g.* a number of points of the 3D mesh. ρ is a non-negative and non-increasing function. In example, ρ is compactly supported. y is the geometrical shape. Formula (1) features squares of a norm of the L^2 norm type, namely the terms of the type $\|g(z)_j - y_j\|_2^2$. It must be understood that such a norm is an example, put in formula (1) for the sake of clarity. The norm may however be replaced by any other norm or distance.

[0084] Alternatively, when the geometrical shape is defined in a 2D plane and the derivative is a projection of the result of applying the decoder to the explored latent vector, the term is of the type:

$$\sum_{s\in S}\min_{j=1,..,M}\|P[g(z)_j] - s\|_2^2 (2).$$

[0085] In formula (2), z is the explored latent vector, M is a sampling size of the 3D modeled object, S is a set of pixel coordinates of the geometrical shape, $g(z)$ is the result of applying the decoder to the explored latent vector and P is the projection. In examples, the user has sketched the deformation constraint by drawing over the 3D modeled object. In these examples, the geometrical shape is thus a set of pixels defined by the user's sketch. P may be a projection from the 3D modeled object space over a 2D space of objects defined with pixel coordinates. Once, more it is to be understood that the norm of the L^2 norm type of formula (2) is an example, put in the formula for the sake of clarity. The norm may however be replaced by any other norm or distance.

[0086] The applying S50 of the decoder to the optimal latent vector is now discussed.

[0087] Applying S50 the decoder to the optimal latent vector outputs a 3D modeled object. The 3D modeled object is inferred from the optimal latent vector. In other words, the method outputs the 3D modeled object corresponding to a minimum of the energy, that is a 3D modeled object which tends to respect (*e.g.* respects, *e.g.* respects best) the deformation constraint. In examples where 3D modeled objects involved in the method are plausible 3D modeled objects, the method thus outputs a plausible 3D modeled object *(e.g.* of a class) which tends to respect *(e.g.* respects, *e.g.* respects best) the deformation constraint among several plausible 3D modeled objects *(e.g.* of the class).

[0088] In examples, the method further comprises computing S60 a deformation operation, from the result of applying

the decoder to the projection of the 3D modeled object onto the latent space, to the result of applying S50 the decoder to the optimized latent vector. In these examples, the method further comprises applying S70 the deformation operation to the 3D modeled object.

[0089] Computing S60 the deformation operation, from the result of applying the decoder to the projection of the 3D modeled object onto the latent space to the result of applying the decoder to the optimized latent vector means finding the deformation operation. The deformation operation transforms the result of applying the decoder to the projection of the 3D modeled object onto the latent space into the result of applying S50 the decoder to the optimized latent vector. In examples where projecting the 3D modeled object onto the latent space means applying an encoder of an autoencoder also comprising the machine-learnt decoder to the 3D modeled object, the method thus computes S60 a deformation which transforms the reconstructed 3D modeled object into the 3D modeled object corresponding to a minimum of the energy, that is a 3D modeled object which tends to respect (*e.g.* respects, *e.g.* respects best) the deformation constraint. Applying S70 the deformation operation to the 3D modeled object thus deforms the 3D modeled object into another 3D modeled object that tends to respect (*e.g.* respects, *e.g.* respects optimally) the deformation constraint. Such a deformation may preserve (*e.g.* topological) details of the 3D modeled object.

[0090] In examples where the 3D modeled objects involved in the method are plausible 3D modeled objects, the method may thus compute a transformation which transforms the 3D modeled object into another 3D modeled object that is plausible and that tends to respect (*e.g.* respects, *e.g.* respects best) the deformation constraint. In other words, upon the providing of the deformation constraint, the method may thus automatically deform the 3D modeled object into a 3D modeled object (*e.g.* of a class) which is plausible, and which tends to respect (*e.g.* respects, *e.g.* respects best) the deformation constraint among several other plausible objects (*e.g.* of the class). Also, deforming the 3D modeled object allows to preserve (*e.g.* topological) details of the 3D modeled object that the decoded optimal latent vector may not feature. In other words, outputting the decoded latent vector by itself may provide a plausible 3D modeled object tending to respect (*e.g.* respecting) the deformation constraint, but outputting in addition a deformation of the 3D modeled object into the decoded latent vector allows to output a plausible 3D object tending to respect (*e.g.* respecting) the deformation constraint as well as topological details of the provided 3D modeled object. The method is thus efficient, accurate and ergonomic.

[0091] Examples of the determining S40 are now discussed.

[0092] In examples, the machine-learnt decoder may be the decoder of an (*e.g.* machine-learnt) autoencoder that also comprises an encoder. In these examples, the determining S40 of the optimal latent vector may reward similarity between the result of applying the decoder to the explored latent vector and the 3D modeled object.

[0093] Rewarding may generally mean that the minimizing of the energy is conceived to tend to a result (*i.e.* the optimal latent vector) which is close to the projection of the 3D modeled object onto the latent space. Rewarding the similarity may comprise exploring latent vectors which are (*e.g.* relatively) close to the projection of the 3D modeled object onto the latent space. For example the exploring of the latent vectors may be performed iteratively (that is, latent vectors are explored one by one, until the optimal latent vector is found) and may start from a first latent vector which is the projection of the 3D modeled object onto the latent space. Alternatively or additionally, rewarding the similarity may be performed by considering an energy that comprises a quantity that is an increasing function of an error between the result of applying the decoder to the explored latent vector and the 3D modeled object.

[0094] Rewarding similarity allows to reward 3D modeled objects (*i.e.* decoded latent vector) which are similar to the provided 3D modeled object. Thus, the decoded optimal latent vector is not only a 3D modeled object which fits the deformation constraint but is also a 3D modeled object which is similar to the provided 3D modeled object. In examples, the method thus allows to determine a plausible 3D modeled object (*e.g.* of a class) which, among other plausible 3D modeled objects (*e.g.* of the class), fits (*e.g.* best) the deformation constraint defined by the user and the shape of the provided 3D modeled object. The method is thus efficient and accurate.

[0095] In these examples, the determining S40 of the optimal latent vector is performed iteratively, starting from a first explored latent vector which is the result of applying the encoder to the 3D modeled object. Starting this first explored latent vector is a particular efficient way to minimize the energy with an optimal latent vector which is relatively close to the encoded 3D modeled object, since exploration of latent vectors is carried out near the encoded 3D modeled object.

[0096] Additionally or alternatively the energy may comprise another term that rewards similarity between the explored latent vector and the result of applying the encoder to the 3D modeled object. Rewarding the similarity between the explored latent vector and the result of applying the encoder to the 3D modeled object allows to avoid exploring latent vectors which are too far from the encoded 3D modeled object.

[0097] In examples, the other term involves a distance between the explored latent vector and the result of applying the encoder to the 3D modeled object. The distance is an efficient way to measure and reward closeness between latent vectors and the encoded 3D modeled object.

[0098] Rewarding the similarity between the explored latent vector and the result of applying the encoder to the 3D modeled object may mean that the other term is an increasing function of the distance between the explored latent vector and the result of applying the encoder to the 3D modeled object.

[0099] In examples, the energy comprises another term which rewards a likelihood of the explored latent vector on the latent space. In examples where a probability prior $p(z)$ is given over the latent space, z being a variable representing latent vectors, the term may measure the largeness of $p(z)$. The term may for example be a decreasing function of $p(z)$. Thus, when $p(z)$ is close to 1, that is when the latent vector has a large likelihood, the term is small. The likelihood of the optimal latent vector, and thus of the decoded optimal latent vector, is thus maximized.

[0100] Two examples of the method are now discussed. In the first example, the geometrical shape is defined in a 3D space and the derivative is the result of applying the decoder to the explored latent vector is now discussed. In the second example, the geometrical shape is defined in a 2D plane and the derivative is a projection of the result of applying the decoder to the explored latent vector. FIG. 4-9 illustrate the first example. FIG. 10-13 illustrate the second example.

[0101] In both examples, the user provides S10 the 3D modeled object which is a 3D mesh x that is a discretization of the provided 3D modeled object. In other words, the provided 3D modeled object is a 3D mesh. It is also provided S10 a machine-learnt autoencoder $g \circ f$, where g is the machine-learnt decoder and where f is a machine-learnt encoder. $g(f(x))$ is the reconstructed mesh representing the provided 3D modeled object, and let M be a notation for its number of points (i.e. its sampling size). In examples, a probabilistic prior $p(z)$ is given over the latent space. In examples where this is not the case, $p(z)$ may simply be replaced by $p(z) = 1$ in all equations where $p(z)$ appears, corresponding to a uniform prior. In examples, the method comprises projecting S20 x onto the latent space, by computing $z_0 = f(x)$.

[0102] In the first example, the mesh x comprises control points $a_1, ..., a_n$ at characteristic places, using 3D interest points to find appropriate places. The user defines S30 a deformation constraint by pulling (e.g. through a drag and drop operation) one or several of these control points denoted $a_1, ..., a_k$ in order to deform the part of the 3D mesh. Let $a'_1, ..., a'_n$ be the coordinates of the control points after defining the deformation constraint. The deformation relative to the deformation constraint (i.e. the deformation resulting from applying the deformation constraint) may be computed with a Radial Basis Function (RBF) interpolation. The method may consider the RBF by a well-chosen function φ. The method

$$\delta(X, Y, Z) = \sum_{i=1}^{n} \begin{pmatrix} r_i \\ s_i \\ t_i \end{pmatrix} \varphi \left(\left\| \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - a_i \right\| \right)$$

may fit a deformation field by solving the three following linear systems for r, s and t:

$$\forall j = 1, ..., n,$$

$$\sum_{i=1}^{n} \begin{pmatrix} r_i \\ s_i \\ t_i \end{pmatrix} \varphi(\|a_j - a_i\|) = a'_j - a_j$$

[0103] FIG. 4 illustrate an example of the mesh x. The mesh represents a plausible chair 40 with four legs, such as one of the chairs of FIG. 3. The chair is meshed, and the mesh comprises a control point a_1. As illustrated in FIG. 5, the user pulls the point a_1 in order to deform a leg of the chair 40 (which is a part of the chair). FIG. 5 shows the control point a_1' after deformation.

[0104] In this first example $v_1, ..., v_k$ be the indices of the closest vertices of the reconstructed 3D mesh $g(z)$ to $a_1, ..., a_k$. In this example, $y = g(z_0) + \delta(g(z_0))$ is the deformed reconstructed 3D mesh under the deformation field δ. In other words, y is the new geometrical shape defined by the deformation constraint. FIG. 6 shows the reconstruction 60 of the 3D meshed chair 40 and FIG. 7 shows the deformed reconstructed 3D mesh 70.

[0105] The optimal latent vector is determined S40 by minimizing the energy, which is of the form:

$$E(z) = \sum_{i=1}^{k} \left[\sum_{j=1}^{M} \rho(\| g(z)_{v_i} - g(z)_j \|) \| g(z)_j - y_j \|_2^2 \right] + \gamma \| z - z_0 \|_2^2$$

$$- \beta \log(p(z))$$

where ρ is non-negative non-increasing function, e.g. with a compact support. ρ is used to fit the reconstruction to the user deformation only around where the user has pulled control points, so as to penalize non-respect of the deformation constraint by $g(z)$. The method may optimize the latent vector z starting from z_0, to modify the reconstructed input $g(z)$ such that it fits the deformation given by the user.

[0106] The energy may comprise another term $\gamma \| z - z_0 \|_2^2$. In examples, γ equals zero such the other term is not comprised in the energy. In other examples, γ is not equal to zero, such that the other term is comprised in the energy. The other term is a smoothness term that is a distance between the explored latent vector z and the result z_0 of applying the encoder to the 3D modeled object. The other term prevents the optimization from mowing z too far away from z_0, and thus remains as close as possible to the original input, while fitting to the user-defined deformation field.

[0107] $-\beta \log(p(z))$ is another term that rewards a likelihood of the explored latent vector z on the latent space, p(z) being a probabilistic prior given over the latent space. The another term is used to maximize the likelihood of the optimized shape. In examples, β equals zero, such that the other term is not comprised in the energy. In other examples, β is strictly positive, such that the other term is comprised in the energy.

[0108] The minimizing of E (*i.e.* the determining of the optimal latent vector) may comprise applying an iteratively reweighted least squares method, meaning that at each optimization iteration the method updates the least squares weights $\rho(\|g(z)_{v_i} - g(z)_j\|)$, but considers them fixed during the optimization step.

[0109] FIG. 8 illustrates the result of applying the decoder to the optimized latent vector in the case of the first example. The chair 80 is a mesh representing a plausible chair that fits the deformation constraint defined by the user. The method may further comprise computing a deformation from the reconstructed chair 60 to the chair 80 fitting the deformation constraint. The result of applying the deformation to the chair 40 is a plausible chair 90 of FIG. 9 that respects both the deformation constraints and the initial shape of chair 40.

[0110] An implementation of the computing and the applying the deformation in the case of the first example is now further discussed.

[0111] The minimizing of the energy according to the first example led to an optimized latent vector $\hat{z} = \text{argmin}_z E(z)$ which is as close as possible to the original input, while fitting to the user constraints, either given by control points or sketch. Nevertheless, this optimized shape is computed starting from the reconstruction $g(z_0)$ (illustrated by chair 80), which is not necessarily an exact reconstruction with perfect accuracy. Especially, the reconstruction does not have necessarily the same topology (e.g. number of chair rungs), and the local details of the input 3D modeled object 40 are not necessarily included in the reconstruction. In order to remain as close as possible to the style of the original chair 40, while fitting to the user constraints, the method may retarget the deformations between $g(z_0)$ (illustrated by chair 80) and $g(\hat{z})$ (illustrated by chair 70) to the original chair x. The method may use RBF interpolation to compute the deformation field θ between $g(z_0)$ and $g(\hat{z})$. The technique may be the same as the previously discussed RBF interpolation, except that the method may not use any control points, but instead may compute the deformation field using directly the vertices of the reconstruction (or a characteristic subset of these vertices).

$$\theta(X, Y, Z) = \sum_{i=1}^{M} \begin{pmatrix} r_i \\ s_i \\ t_i \end{pmatrix} \varphi\left(\left\| \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - g(z_0)_i \right\|\right),$$

[0112] Thus, to derive , the method may solve the linear system

$$\forall j = 1, \ldots, M,$$

$$\sum_{i=1}^{M} \begin{pmatrix} r_i \\ s_i \\ t_i \end{pmatrix} \varphi(\|g(z_0)_j - g(z_0)_i\|) = g(\hat{z})_j - g(z_0)_j$$

[0113] The final edited mesh 90 is simply given by the deformation field applied to the original mesh x:

$$x + \theta(x)$$

[0114] Notice that other retargeting methods could also be used to compute the deformation field, such that non rigid registration (See "Global Correspondence Optimization for Non-Rigid Registration of Depth Scans." H. Li, R. Sumner, and M. Pauly, in Comput. Graph. Forum 27 (5): 1421-1430, 2008), or kernel regression.

[0115] In the second example, the user sketches over the 3D modeled object x. in a chosen view. FIG. 10 shows an example of the 3D modeled object x, which is a chair 100 with four legs such as the ones of FIG. 10. FIG. 10 shows a front view of the chair 100. Let P be the projection associated to the chosen view. P may be the projection from the 3D modeled object space to which the chair belongs and onto the 2D space of pixel coordinates on the screen. The user defines the deformation constraint by sketching a shape 102 on the screen. Let S be the pixels coordinates of the drawn sketch, or sub-sampled version of this drawing.

[0116] The optimal latent vector is determined S40 by minimizing the energy, which is of the form:

$$E(z) = \sum_{s \in S} \min_{j=1,..,M} \| P[g(z)_j] - s \|_2^2 + \gamma \| z - z_0 \|_2^2 - \beta \log(p(z))$$

[0117] The method may optimize the latent vector z starting from z_0, to modify the reconstructed input $g(z)$ such that it fits the sketch given by the user.

[0118] The energy may comprise another term $\gamma \| z - z_0 \|_2^2$. In examples, γ equals zero such the other term is not comprised in the energy. In other examples, γ is not equal to zero, such that the other term is comprised in the energy. The other term is a smoothness term that is a distance between the explored latent vector z and the result z_0 of applying the encoder to the 3D modeled object. The other term prevents the optimization from mowing z too far away from z_0, and thus remains as close as possible to the original input, while fitting to the user-defined deformation field.

[0119] $- \beta \log(p(z))$ is another term that rewards a likelihood of the explored latent vector z on the latent space, $p(z)$ being a probabilistic prior given over the latent space.. The another term is used to maximize the likelihood of the optimized shape. In examples, β equals zero, such that the other term is not comprised in the energy. In other examples, β is strictly positive, such that the other term is not comprised in the energy.

[0120] FIG. 12 shows an example of the result of applying S50 the decoder to the optimized latent vector. The chair 120 is a mesh representing a plausible chair that fits the deformation constraint defined by the user. The method may further comprise computing S60 a deformation from the reconstructed chair 110 of FIG. 11 to the chair 120 fitting the deformation constraint. The result of applying S70 the deformation to the chair 100 is a plausible chair 130 that respects both the deformation constraints and the initial shape of chair 100.

[0121] An implementation of the computing S60 and the applying S70 the deformation in the case of the second example is now further discussed.

[0122] The minimizing of the energy according to the second example led to an optimized latent vector $\hat{z} = \text{argmin}_z E(z)$ which is as close as possible to the original input, while fitting to the user constraints, either given by control points or sketch. Nevertheless, this optimized shape is computed starting from the reconstruction $g(z_0)$ (illustrated by chair 100), which is not necessarily an exact reconstruction with perfect accuracy. Especially, the reconstruction does not have necessarily the same topology (e.g. number of chair rungs), and the local details of the input 3D modeled object 100 are not necessarily included in the reconstruction. In order to remain as close as possible to the style of the original chair 100, while fitting to the user constraints, the method may retarget the deformations between $g(z_0)$ (illustrated by chair 100) and $g(\hat{z})$ (illustrated by chair 120) to the original chair x. The method may use RBF interpolation to compute the deformation field θ between $g(z_0)$ and $g(\hat{z})$. The technique may be the same as the previously discussed RBF interpolation, except that the method may not use any control points, but instead may compute the deformation field using directly the vertices of the reconstruction (or a characteristic subset of these vertices).

$$\theta(X, Y, Z) = \sum_{i=1}^{M} \begin{pmatrix} r_i \\ s_i \\ t_i \end{pmatrix} \varphi \left(\left\| \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} - g(z_0)_i \right\| \right)$$

[0123] Thus, to derive ⟨above⟩, the method may solve the linear system

$$\forall j = 1, \ldots, M,$$

$$\sum_{i=1}^{M} \begin{pmatrix} r_i \\ s_i \\ t_i \end{pmatrix} \varphi \left(\| g(z_0)_j - g(z_0)_i \| \right) = g(\hat{z})_j - g(z_0)_j$$

[0124] The final edited mesh 130 is simply given by the deformation field applied to the original mesh x:

$$x + \theta(x)$$

[0125] Notice that other retargeting methods could also be used to compute the deformation field, such that non rigid registration (See "Global Correspondence Optimization for Non-Rigid Registration of Depth Scans." H. Li, R. Sumner, and M. Pauly, in Comput. Graph. Forum 27 (5): 1421-1430, 2008), or kernel regression.

[0126] FIG. 14 shows an example of the GUI of the system, wherein the system is a CAD system.

[0127] The GUI 2100 may be a typical CAD-like interface, having standard menu bars 2110, 2120, as well as bottom and side toolbars 2140, 2150. Such menu- and toolbars contain a set of user-selectable icons, each icon being associated with one or more operations or functions, as known in the art. Some of these icons are associated with software tools, adapted for editing and/or working on the 3D modeled object 2000 displayed in the GUI 2100. The software tools may be grouped into workbenches. Each workbench comprises a subset of software tools. In particular, one of the workbenches is an edition workbench, suitable for editing geometrical features of the modeled product 2000. In operation, a designer may for example pre-select a part of the object 2000 and then initiate an operation (e.g. change the dimension, color, etc.) or edit geometrical constraints by selecting an appropriate icon. For example, typical CAD operations are the modeling of the punching or the folding of the 3D modeled object displayed on the screen. The GUI may for example display data 2500 related to the displayed product 2000. In the example of the figure, the data 2500, displayed as a "feature tree", and their 3D representation 2000 pertain to a brake assembly including brake caliper and disc. The GUI may further show various types of graphic tools 2130, 2070, 2080 for example for facilitating 3D orientation of the object, for triggering a simulation of an operation of an edited product or render various attributes of the displayed product 2000. A cursor 2060 may be controlled by a haptic device to allow the user to interact with the graphic tools.

[0128] FIG. 15 shows an example of the system, wherein the system is a client computer system, *e.g.* a workstation of a user.

[0129] The client computer of the example comprises a central processing unit (CPU) 1010 connected to an internal communication BUS 1000, a random access memory (RAM) 1070 also connected to the BUS. The client computer is further provided with a graphical processing unit (GPU) 1110 which is associated with a video random access memory 1100 connected to the BUS. Video RAM 1100 is also known in the art as frame buffer. A mass storage device controller 1020 manages accesses to a mass memory device, such as hard drive 1030. Mass memory devices suitable for tangibly embodying computer program instructions and data include all forms of nonvolatile memory, including by way of example semiconductor memory devices, such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks 1040. Any of the foregoing may be supplemented by, or incorporated in, specially designed ASICs (application-specific integrated circuits). A network adapter 1050 manages accesses to a network 1060. The client computer may also include a haptic device 1090 such as cursor control device, a keyboard or the like. A cursor control device is used in the client computer to permit the user to selectively position a cursor at any desired location on display 1080. In addition, the cursor control device allows the user to select various commands, and input control signals. The cursor control device includes a number of signal generation devices for input control signals to system. Typically, a cursor control device may be a mouse, the button of the mouse being used to generate the signals. Alternatively or additionally, the client computer system may comprise a sensitive pad, and/or a sensitive screen.

[0130] The computer program may comprise instructions executable by a computer, the instructions comprising means for causing the above system to perform the method. The program may be recordable on any data storage medium, including the memory of the system. The program may for example be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. The program may be implemented as an apparatus, for example a product tangibly embodied in a machine-readable storage device for execution by a programmable processor. Method steps may be performed by a programmable processor executing a program of instructions to perform functions of the method by operating on input data and generating output. The processor may thus be programmable and coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. The application program may be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired. In any case, the language may be a compiled or interpreted language. The program may be a full installation program or an update program. Application of the program on the system results in any case in instructions for performing the method.

Claims

1. A computer-implemented method for designing a 3D modeled object via graphical user-interaction with a displayed representation of the 3D modeled object, the method comprising:

   • providing (S10):

     ▪ the 3D modeled object; and
     ▪ a machine-learnt decoder which is a differentiable function taking values in a latent space and outputting values in a 3D modeled object space;

   • defining (S30), by graphical user interaction with the displayed representation of the 3D modeled object, a

deformed geometrical shape of the 3D modeled object, the deformed geometrical shape defining a deformation constraint for a part of the 3D modeled object;
• determining (S40) an optimal latent vector which minimizes an energy, the energy exploring latent vectors and comprising a term which penalizes, for each explored latent vector, non-respect of the deformation constraint by the result of applying the decoder to the explored latent vector; and
• applying (S50) the decoder to the optimal latent vector,
wherein the deformation constraint defines a geometrical shape and the term penalizes a disparity between the result of applying the decoder to the explored latent vector and the geometrical shape,
wherein the disparity involves a distance between a derivative of the result of applying the decoder to the explored latent vector and the geometrical shape,
wherein:

> • the geometrical shape is defined in a 3D space and the derivative is the result of applying the decoder to the explored latent vector; or
> • the geometrical shape is defined in a 2D plane and the derivative is a projection of the result of applying the decoder to the explored latent vector,

wherein the disparity is a monotonic function of the distance,
wherein the 3D modeled object is a 3D mesh and:

> • when the geometrical shape is defined in a 3D space and the derivative is the result of applying the decoder to the explored latent vector, the term is of the type:

$$\sum_{i=1}^{k} \left[\sum_{j=1}^{M} \rho \left(\| g(z)_{v_i} - g(z)_j \| \right) \| g(z)_j - y_j \|_2^2 \right],$$

> where

> - z is the explored latent vector;
> - $g(z)$ is the result of applying the decoder to the explored latent vector;
> - $v_1, v_2, \ldots, v_k$ are the indices of k vertices of $g(z)$ representing the part of the 3D modeled object;
> - M is a sampling size of the 3D modeled object;
> - ρ is a non-negative and non-increasing function;
> - γ is the geometrical shape; or

> • when the geometrical shape is defined in a 2D plane and the derivative is a projection of the result of applying the decoder to the explored latent vector, the term is of the type:

$$\sum_{s \in S} \min_{j=1,..,M} \| P[g(z)_j] - s \|_2^2,$$

> where

> - z is the explored latent vector;
> - M is a sampling size of the 3D modeled object;
> - S is a set of pixel coordinates of the geometrical shape;
> - $g(z)$ is the result of applying the decoder to the explored latent vector;
> - P is the projection,

wherein:

> • when the geometrical shape is defined in a 3D space, the graphical user interaction which defines the deformation constraint consists in defining the geometrical shape by clicking or touching a portion of the displayed 3D modeled object and deforming said portion with a drag and drop operation, or
> • when the geometrical shape is defined in a 2D plane, the graphical user interaction which defines the deformation constraint consists in defining the geometrical shape by graphically drawing a 2D drawing that forms the geometrical shape.

2. The method of claim 1, further comprising:

• computing (S60) a deformation operation, from the result of applying the decoder to the projection of the 3D modeled object onto the latent space, to the result of applying (S50) the decoder to the optimized latent vector; and
• applying (S70) the deformation operation to the 3D modeled object.

3. The method of any one of claims 1 to 2, wherein the machine-learnt decoder is the decoder of an autoencoder that also comprises an encoder, and wherein the determining (S40) of the optimal latent vector rewards similarity between the result of applying the decoder to the explored latent vector and the 3D modeled object.

4. The method of claim 3, wherein the determining (S40) of the optimal latent vector is performed iteratively, starting from a first explored latent vector which is the result of applying the encoder to the 3D modeled object.

5. The method of claim 3 or 4, wherein the energy comprises another term that rewards similarity between the explored latent vector and the result of applying the encoder to the 3D modeled object.

6. The method of claim 5, wherein the other term involves a distance between the explored latent vector and the result of applying the encoder to the 3D modeled object.

7. The method of any one of claims 1 to 6, wherein the energy comprises another term which rewards a likelihood of the explored latent vector on the latent space.

8. A computer program comprising instructions which, when the program is executed by a computer, cause the computer to carry out the method according to any one of the claims 1 to 7.

9. A computer readable storage medium having recorded thereon the computer program of claim 8.

10. A computer comprising a processor coupled to a memory and a display, the memory having recorded thereon the computer program of claim 8.

Patentansprüche

1. Computerimplementiertes Verfahren zum Konstruieren eines modellierten 3D-Objekts über eine grafische Benutzerinteraktion mit einer angezeigten Darstellung des modellierten 3D-Objekts, das Verfahren umfassend:

• Bereitstellen (S10):

▪ des modellierten 3D-Objekts; und
▪ eines maschinell gelernten Decoders, der eine differenzierbare Funktion ist, die Werte in einem latenten Raum annimmt und Werte in einem modellierten 3D-Objektraum ausgibt;

• Definieren (S30), durch grafische Benutzerinteraktion mit der angezeigten Darstellung des modellierten 3D-Objekts, einer deformierten geometrischen Form des modellierten 3D-Objekts, wobei die deformierte geometrische Form eine Deformationsbeschränkung für einen Teil des modellierten 3D-Objekts definiert;
• Bestimmen (S40) eines optimalen latenten Vektors, der eine Energie minimiert, wobei die Energie latente Vektoren erkundet und einen Term umfasst, der für jeden erkundeten latenten Vektor eine Nichteinhaltung der Deformationsbeschränkung durch das Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor bestraft; und
• Anwenden (S50) des Decoders auf den optimalen latenten Vektor,
wobei die Deformationsbeschränkung eine geometrische Form definiert und der Term eine Diskrepanz zwischen dem Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor und der geometrischen Form bestraft,
wobei die Disparität einen Abstand zwischen einer Ableitung des Resultats eines Anwendens des Decoders auf den erkundeten latenten Vektor und der geometrischen Form involviert,
wobei:

• die geometrische Form in einem 3D-Raum definiert ist und die Ableitung ist das Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor ist; oder
• die geometrische Form in einer 2D-Ebene definiert ist und die Ableitung ist eine Projektion des Resultats eines Anwendens des Decoders auf den erkundeten latenten Vektor ist,

wobei die Disparität eine monotone Funktion des Abstands ist,
wobei das modellierte 3D-Objekt ein 3D-Netz ist und:

• wenn die geometrische Form in einem 3D-Raum definiert ist und die Ableitung das Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor ist, der Term folgender Art ist:

$$\sum_{i=1}^{k} \left[\sum_{j=1}^{M} \rho \left(\left\| \, g(z)_{v_i} - g(z)_j \, \right\| \right) \left\| \, g(z)_j - y_j \, \right\|_2^2 \right],$$

wobei:

- z der erkundete latente Vektor ist;
- $g(z)$ das Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor ist;
- $v_1, v_2, ..., v_k$ die Indizes von k Scheitelpunkten von $g(z)$ sind, die den Teil des modellierten 3D-Objekts darstellen;
- M der Stichprobenumfang des modellierten 3D-Objekts ist;
- ρ eine nicht-negative und nicht-erhöhende Funktion ist;
- y die geometrische Form ist; oder

• wenn die geometrische Form in einer 2D-Ebene definiert ist und die Ableitung eine Projektion des Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor ist, der Term folgender Art ist:

$$\sum_{s \in S} \min_{i=1,...,M} \left\| \, P[g(z)_j] - s \, \right\|_2^2,$$

wobei:

- z der erkundete latente Vektor ist;
- M der Stichprobenumfang des modellierten 3D-Objekts ist;
- S ein Satz von Pixelkoordinaten der geometrischen Form ist;
- $g(z)$ das Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor ist;
- P die Projektion ist,

wobei:

• wenn die geometrische Form in einem 3D-Raum definiert ist, die grafische Benutzerinteraktion, die die Deformationsbeschränkung definiert, darin besteht, die geometrische Form durch Anklicken oder Berühren eines Teils des angezeigten modellierten 3D-Objekts zu definieren und diesen Teil durch Ziehen und Ablegen zu deformieren, oder
• wenn die geometrische Form in einer 2D-Ebene definiert ist, die grafische Benutzerinteraktion, die die Deformationsbeschränkung definiert, darin besteht, die geometrische Form durch grafisches Zeichnen einer 2D-Zeichnung, die die geometrische Form bildet, zu definieren.

2. Verfahren nach Anspruch 1, ferner umfassend:

• Berechnen (S60) einer Deformationsoperation aus dem Resultat eines Anwendens des Decoders auf die Projektion des modellierten 3D-Objekts auf den latenten Raum und dem Resultat eines Anwendens (S50) des Decoders auf den optimierten latenten Vektor; und
• Anwenden (S70) der Deformationsoperation auf das modellierte 3D-Objekt.

3. Verfahren nach einem der Ansprüche 1 bis 2, wobei der maschinell gelernte Decoder der Decoder eines Auto-

encoders ist, der auch einen Encoder umfasst, und wobei das Bestimmen (S40) des optimalen latenten Vektors die Ähnlichkeit zwischen dem Resultat eines Anwendens des Decoders auf den erkundeten latenten Vektor und dem modellierten 3D-Objekt belohnt.

4. Verfahren nach Anspruch 3, wobei das Bestimmen (S40) des optimalen latenten Vektors iterativ durchgeführt wird, ausgehend von einem ersten erkundeten latenten Vektor, der das Resultat eines Anwendens des Encoders auf das modellierte 3D-Objekt ist.

5. Verfahren nach Anspruch 3 oder 4, wobei die Energie einen anderen Term umfasst, der die Ähnlichkeit zwischen dem erkundeten latenten Vektor und dem Resultat eines Anwendens des Encoders auf das modellierte 3D-Objekt belohnt.

6. Verfahren nach Anspruch 5, wobei der andere Term einen Abstand zwischen dem erkundeten latenten Vektor und dem Resultat eines Anwendens des Encoders auf das modellierte 3D-Objekt involviert.

7. Verfahren nach einem der Ansprüche 1 bis 6, wobei die Energie einen anderen Term umfasst, der eine Wahrscheinlichkeit des erkundeten latenten Vektors auf dem latenten Raum belohnt.

8. Rechnerprogramm, umfassend Anweisungen, die, wenn das Programm von einem Rechner ausgeführt wird, den Rechner veranlassen, das Verfahren nach einem der Ansprüche 1 bis 7 durchzuführen.

9. Rechnerlesbares Speichermedium, auf dem das Rechnerprogramm nach Anspruch 8 aufgezeichnet ist.

10. Rechner, umfassend einen Prozessor, der mit einem Speicher und einer Anzeige gekoppelt ist, wobei in dem Speicher das Rechnerprogramm nach Anspruch 8 aufgezeichnet ist.

Revendications

1. Procédé mis en œuvre par ordinateur pour concevoir un objet modélisé en 3D par le biais d'une interaction utilisateur graphique avec une représentation affichée de l'objet modélisé en 3D, le procédé comprenant le fait de :

• fournir (S10) :

▪ l'objet modélisé en 3D ; et
▪ un décodeur appris par apprentissage automatique qui est une fonction différentiable prenant des valeurs dans un espace latent et fournissant en sortie des valeurs dans un espace d'objets modélisés en 3D ;

• définir (S30), par une interaction utilisateur graphique avec la représentation affichée de l'objet modélisé en 3D, une forme géométrique déformée de l'objet modélisé en 3D, la forme géométrique déformée définissant une contrainte de déformation pour une partie de l'objet modélisé en 3D ;
• déterminer (S40) un vecteur latent optimal qui minimise une énergie, l'énergie explorant les vecteurs latents et comprenant un terme qui pénalise, pour chaque vecteur latent exploré, le non-respect de la contrainte de déformation par le résultat de l'application du décodeur au vecteur latent exploré ; et
• appliquer (S50) le décodeur au vecteur latent optimal,
dans lequel la contrainte de déformation définit une forme géométrique et le terme pénalise une disparité entre le résultat de l'application du décodeur au vecteur latent exploré et la forme géométrique,
dans lequel la disparité implique une distance entre une dérivée du résultat de l'application du décodeur au vecteur latent exploré et la forme géométrique, dans laquelle :

• la forme géométrique est définie dans un espace 3D et la dérivée est le résultat de l'application du décodeur au vecteur latent exploré ; ou
• la forme géométrique est définie dans un plan 2D et la dérivée est une projection du résultat de l'application du décodeur au vecteur latent exploré,

dans lequel la disparité est une fonction monotone de la distance,
dans lequel l'objet modélisé en 3D est un maillage 3D et :

• lorsque la forme géométrique est définie dans un espace 3D et que la dérivée est le résultat de l'application

du décodeur au vecteur latent exploré, le terme est du type :

$$\sum_{i=1}^{k} \left[\sum_{j=1}^{M} \rho\left(\parallel g(z)_{v_i} - g(z)_j \parallel\right) \parallel g(z)_j - y_j \parallel_2^2 \right],$$

où

- z est le vecteur latent exploré ;
- $g(z)$ est le résultat de l'application du décodeur au vecteur latent exploré ;
- v_1, v_2, ..., v_k sont les indices de k sommets de $g(z)$ représentant la partie de l'objet modélisé en 3D ;
- M est une taille d'échantillonnage de l'objet modélisé en 3D ;
- p est une fonction non négative et non croissante ;
- y est la forme géométrique ; ou

• lorsque la forme géométrique est définie dans un plan 2D et que la dérivée est une projection du résultat de l'application du décodeur au vecteur latent exploré, le terme est du type :

$$\sum_{s \in S} \min_{j=1,..,M} \parallel P[g(z)_j] - s \parallel_2^2,$$

où

- z est le vecteur latent exploré ;
- M est une taille d'échantillonnage de l'objet modélisé en 3D ;
- S est un ensemble de coordonnées de pixels de la forme géométrique ;
- $g(z)$ est le résultat de l'application du décodeur au vecteur latent exploré ; ▪ P est la projection,

dans lequel :

• lorsque la forme géométrique est définie dans un espace 3D, l'interaction utilisateur graphique qui définit la contrainte de déformation consiste à définir la forme géométrique en cliquant ou en effleurant une partie de l'objet modélisé 3D affiché et en déformant ladite partie par une opération de glisser-déposer, ou
• lorsque la forme géométrique est définie dans un plan 2D, l'interaction utilisateur graphique qui définit la contrainte de déformation consiste à définir la forme géométrique en dessinant graphiquement un dessin 2D qui forme la forme géométrique.

2. Procédé selon la revendication 1, comprenant en outre le fait de :

• calculer (S60) une opération de déformation, du résultat de l'application du décodeur à la projection de l'objet modélisé en 3D sur l'espace latent, au résultat de l'application (S50) du décodeur au vecteur latent optimisé ; et
• appliquer (S70) l'opération de déformation à l'objet modélisé en 3D.

3. Procédé selon l'une quelconque des revendications 1 à 2, dans lequel le décodeur appris par apprentissage automatique est le décodeur d'un auto-encodeur qui comprend également un encodeur, et dans lequel la détermination (S40) du vecteur latent optimal récompense la similarité entre le résultat de l'application du décodeur au vecteur latent exploré et l'objet modélisé en 3D.

4. Procédé selon la revendication 3, dans lequel la détermination (S40) du vecteur latent optimal est effectuée itérativement, à partir d'un premier vecteur latent exploré qui est le résultat de l'application de l'encodeur à l'objet modélisé en 3D.

5. Procédé selon la revendication 3 ou 4, dans lequel l'énergie comprend un autre terme qui récompense la similarité entre le vecteur latent exploré et le résultat de l'application du codeur à l'objet modélisé en 3D.

6. Procédé selon la revendication 5, dans lequel l'autre terme implique une distance entre le vecteur latent exploré et le résultat de l'application de l'encodeur à l'objet modélisé en 3D.

7. Procédé selon l'une quelconque des revendications 1 à 6, dans lequel l'énergie comprend un autre terme qui récompense une vraisemblance du vecteur latent exploré sur l'espace latent.

8. Programme informatique comprenant des instructions qui, lorsque le programme est exécuté par un ordinateur, amènent l'ordinateur à mettre en œuvre le procédé selon l'une quelconque des revendications 1 à 7.

9. Support de stockage lisible par ordinateur sur lequel est enregistré le programme informatique selon la revendication 8.

10. Ordinateur comprenant un processeur couplé à une mémoire et à un écran, sur laquelle mémoire est enregistré le programme informatique selon la revendication 8.

| providing the 3D modeled object and a machine-learnt decoder | S10 |

| projecting the 3D modeled object onto the latent space | S20 |

| defining, by a user, a deformation constraint | S30 |

| determining an optimal latent vector | S40 |

| applying the decoder to the optimal latent vector | S50 |

| computing a deformation operation | S60 |

| applying the deformation operation | S70 |

FIG. 1

FIG. 2

FIG. 3

40

a_1

FIG. 4

40

$a_1{'}$

FIG. 5

60

FIG. 6

70

FIG. 7

80

FIG. 8

90

FIG. 9

102

100

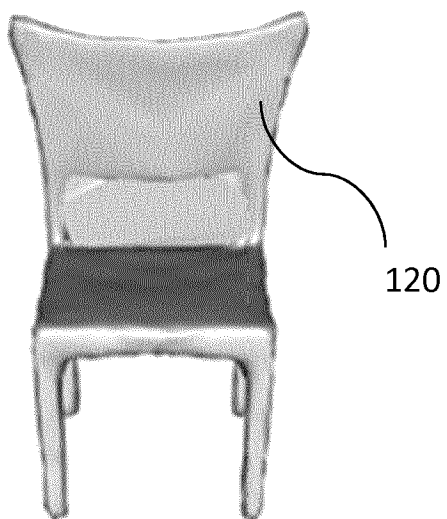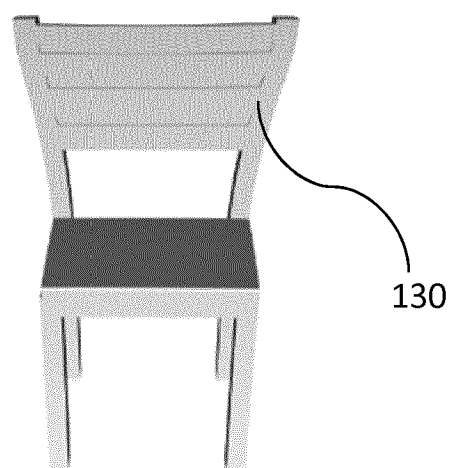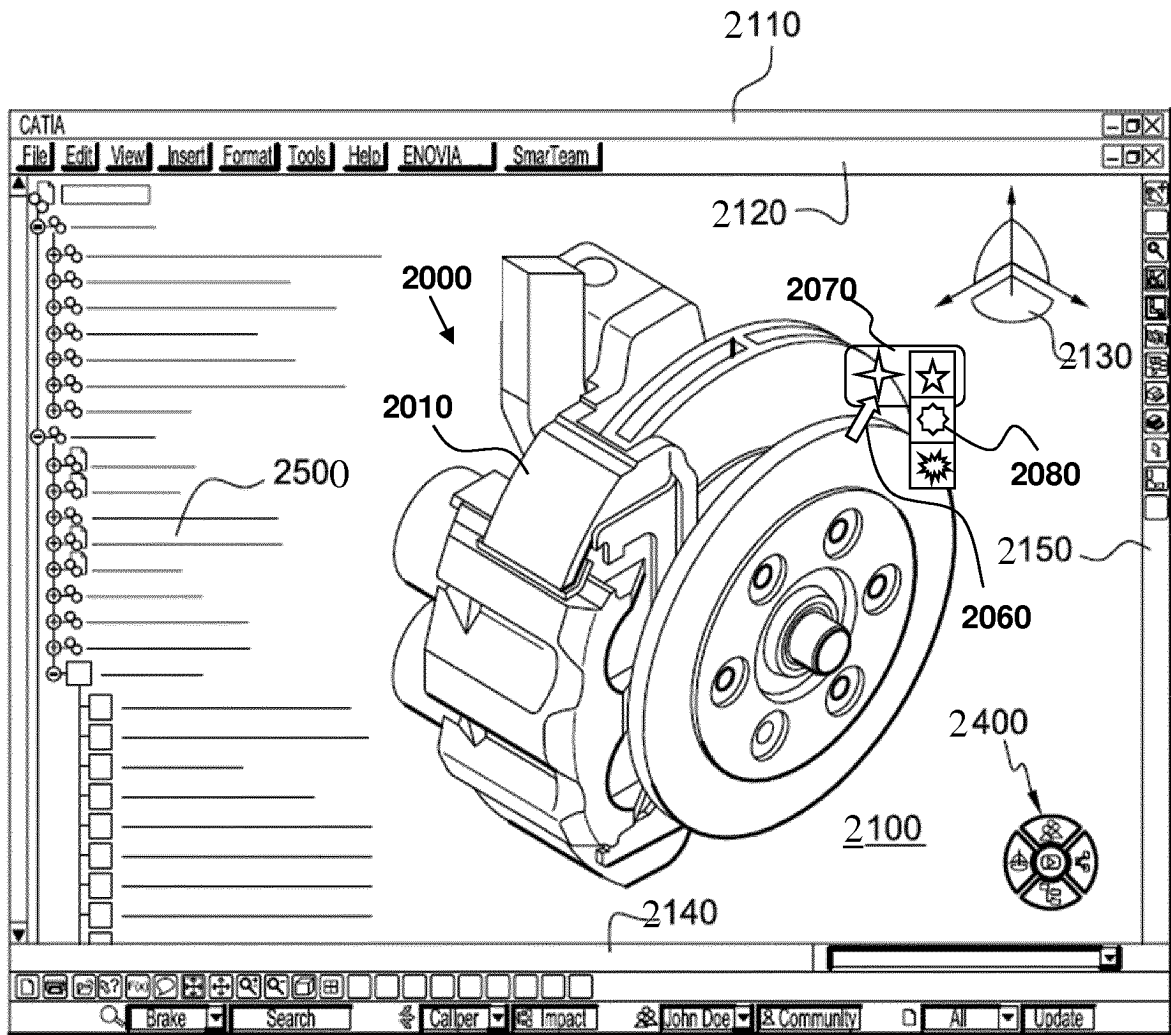FIG. 10

110

FIG. 11

FIG. 12
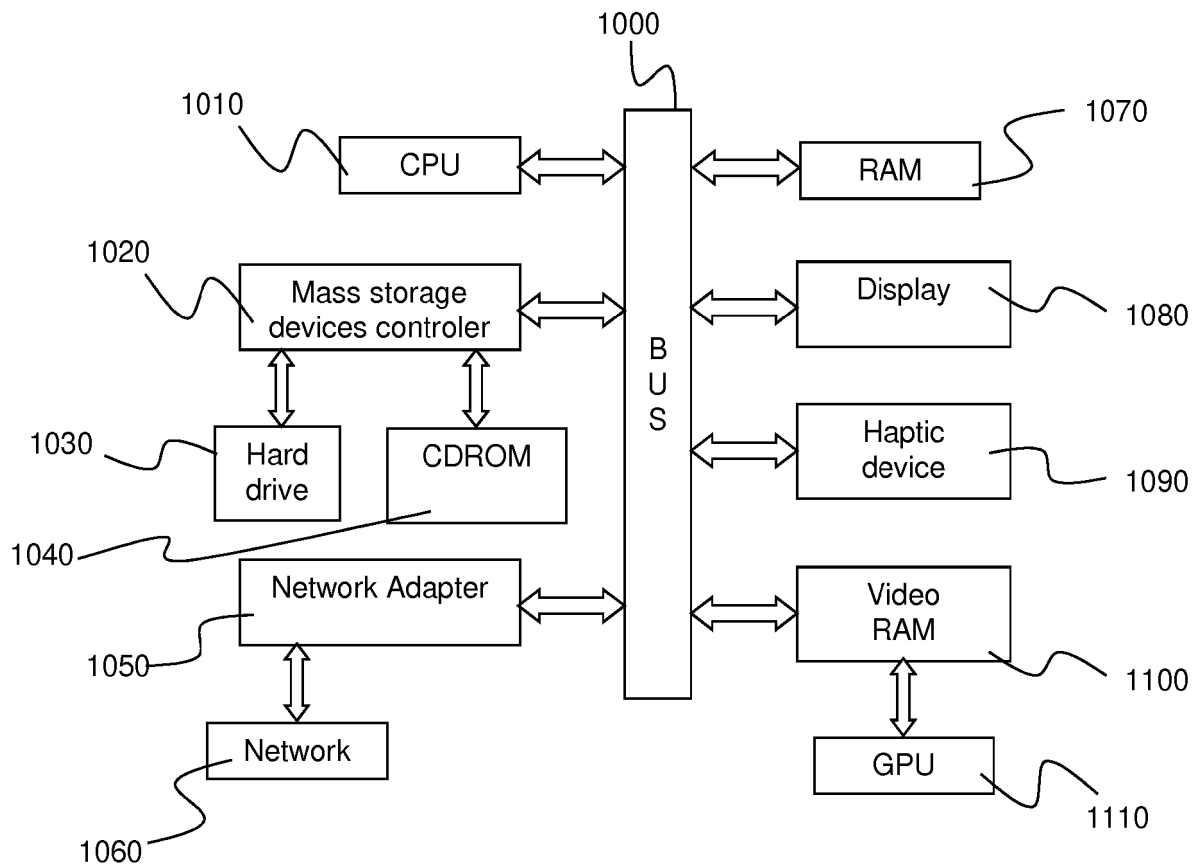
120

FIG. 13

130
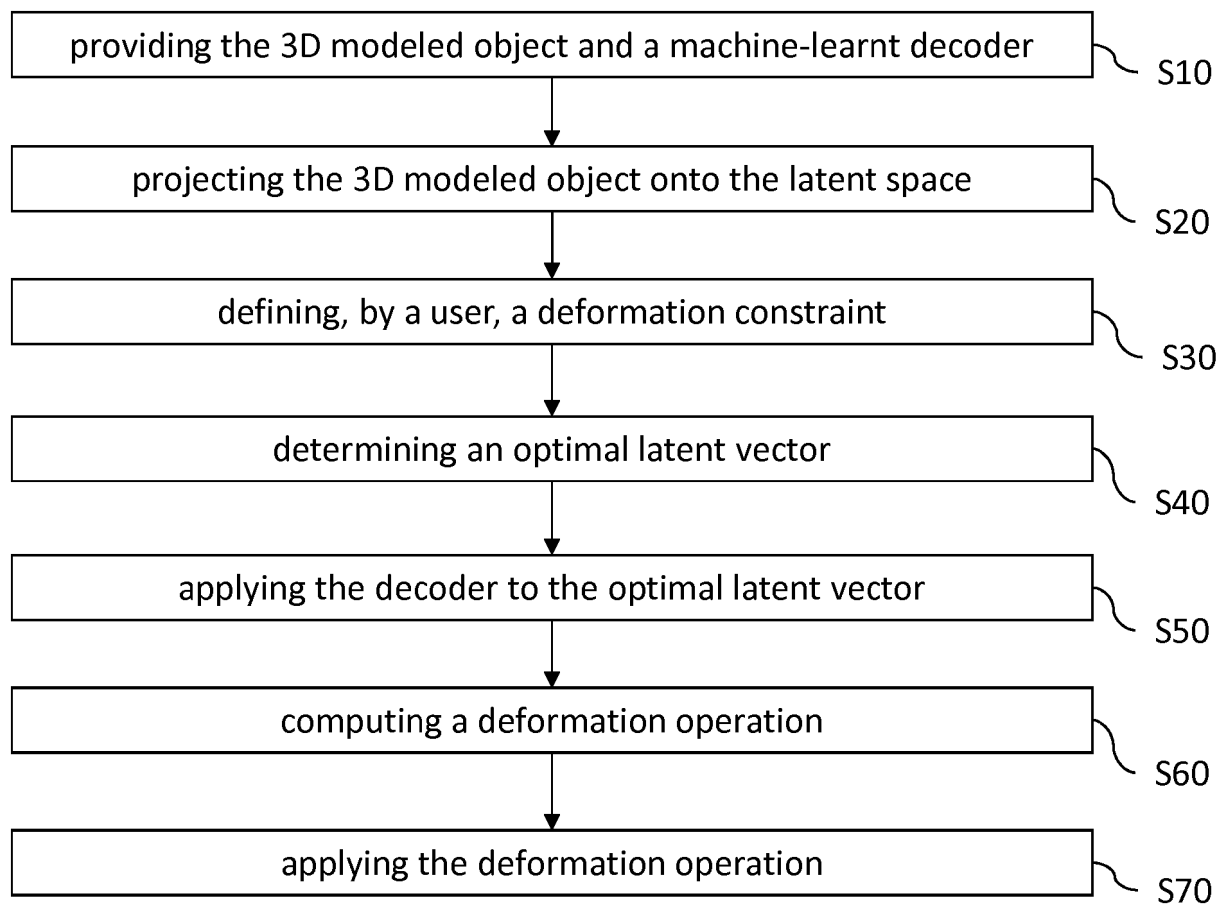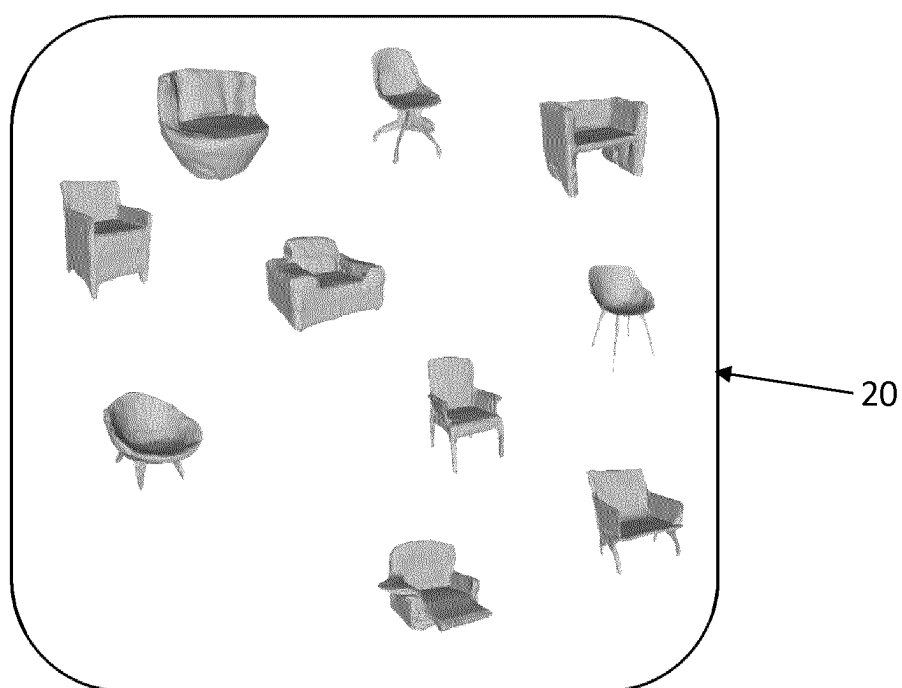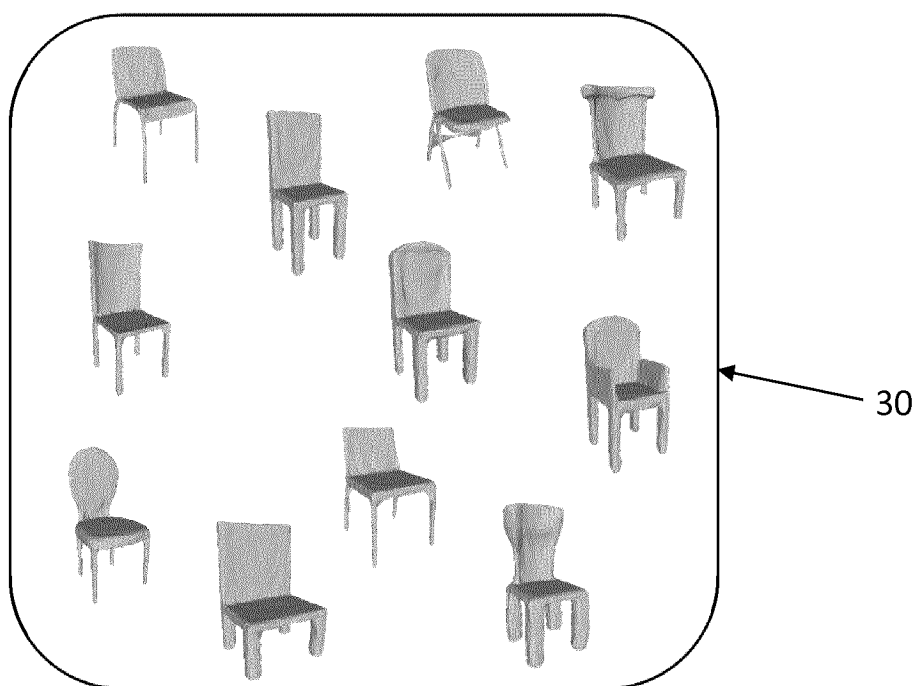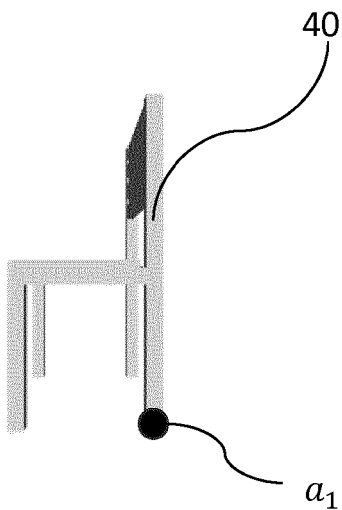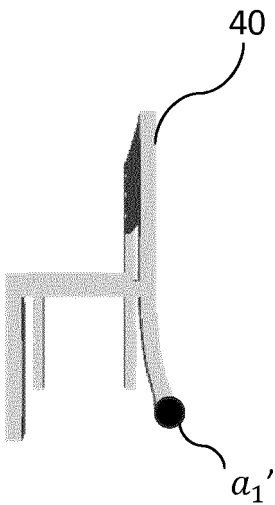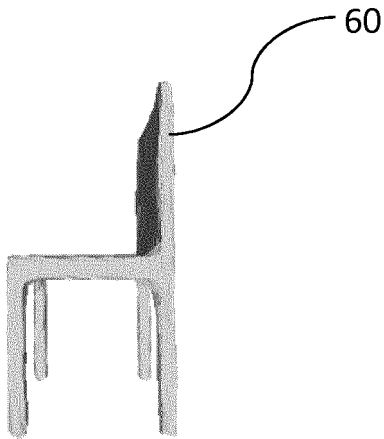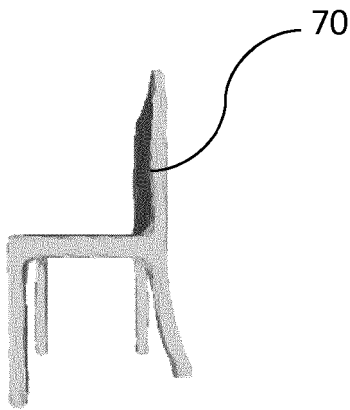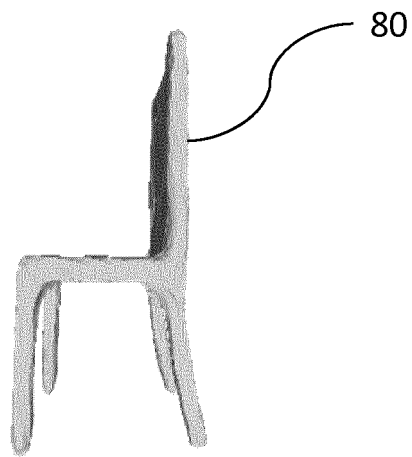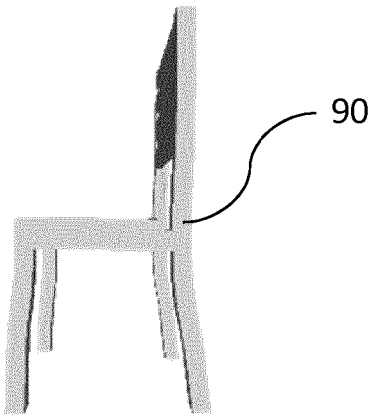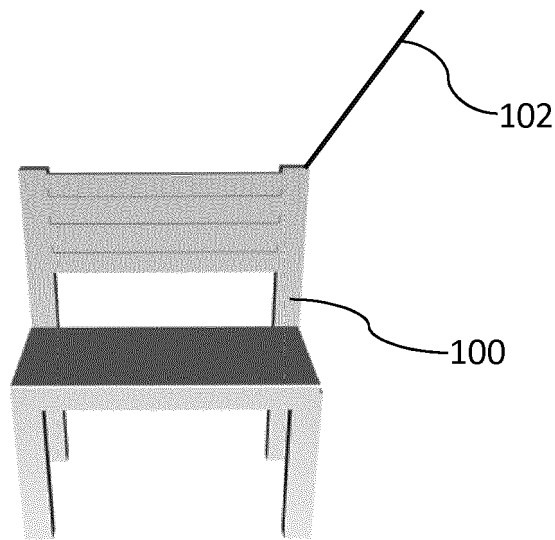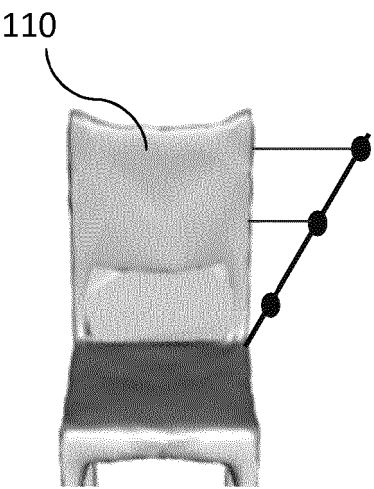
FIG. 14

FIG. 15

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- EP 3396603 A1 **[0009]**

Non-patent literature cited in the description

- **P. VINCENT** ; **H. LARCOHELLE** ; **I. LAJOIE** ; **Y. BENGIO** ; **P. MANZAGOL**. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *The Journal of Machine Learning Research*, 2010 **[0004]**
- **G.E. HINTON** ; **R.R. SALAKHUTDINOV**. Reducing the Dimensionality of Data with Neural Networks. *Science*, 2006 **[0004]**
- **Y. BENGIO**. Learning Deep Architectures for AI. *Foundations and Trends in Machine Learning*, 2009 **[0004]**
- **I. GOODFELLOW** ; **J. POUGET-ABADIE** ; **M. MIRZA** ; **B. XU** ; **D. WARDE-FARLEY** ; **S. OZAIR** ; **A. COURVILLE** ; **Y. BENGIO**. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 2014 **[0004]**
- **J-Y ZHU** ; **P. KRÄHENBÜHL** ; **E. SHECHTMAN** ; **A. A. EFROS**. Generative Visual Manipulation on the Natural Image Manifold. *ECCV*, 2016 **[0004]**
- **A. BROCK** ; **T. LIM** ; **J.M. RITCHIE**. Neural Photo Editing with Introspective Adversarial Networks. *ICLR*, 2017 **[0004]**
- **Q. TAN**. Variational Autoencoders for Deforming 3D Mesh Models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018 **[0010]**
- Learning Semantic Deformation Flows with 3D Convolutional Networks. **E. YUMER et al.** ECCV 2016, Part VI, LNCS 9910. Springer International Publishing AG, 2016, 294-311 **[0011]**
- **J. LIU**. Interactive 3D Modeling with a Generative Adversarial Network. *arXiv:1706.05170v1*, 16 June 2017 **[0012]**
- **Q. TAN**. Mesh-based Autoencoders for Localized Deformation Component Analysis. *arXiv:1709.04304v1*, 13 September 2017 **[0013]**
- **J-Y. ZHU**. Generative Visual Manipulation on the Natural Image Manifold. *arXiv:1609.03552V3*, 16 December 2018 **[0014]**
- **H. LI, R. SUMNER** ; **M. PAULY**. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Comput. Graph. Forum*, 2008, vol. 27 (5), 1421-1430 **[0114] [0125]**